Journal of Harbin Engineering University

ISSN: 1006-7043

Vol 46 No. 7

July 2025

Ensemble of Customized Random Forest and Legacy Recurrent Neural Network with Fuzzy Learning Method Selector

[1]Priya S , [2]Prabhu T

[1]Research Scholar, Dr.M.G.R. Educational and Research Institute,

Chennai, Tamil Nadu, India.

[2]Professor, Department of Computer Applications,

Dr.M.G.R. Educational and Research Institute,

Chennai, Tamil Nadu, India.

Abstract: The rapid integration of Learning Management Systems (LMS) and Learning Analytics Dashboards (LAD) into digital education has transformed instructional delivery through scalable and data-driven platforms. However, these systems often face challenges such as rigid learning pathways, lack of personalization, and limited adaptability to diverse learner behaviors. To overcome these limitations, this work investigates and compares popular machine learning and neural network models in the LMS context. Experimental evaluations reveal that among traditional classifiers, Random Forest (RF) delivers superior performance, while Recurrent Neural Network (RNN) outperforms other neural models. Building on these findings, an intelligent ensemble framework is invented that comprises a Customized Random Forest (CRF) enhanced for contextual learning, a Legacy Recurrent Neural Network (LRNN) refined for temporal pattern recognition, and a Fuzzy Learning Method Selector (FLMS) that dynamically fuses both models based on fuzzy logic rules. The architecture is evaluated using key learning analytics parameters such as Accuracy, Precision, Sensitivity, Specificity, and F-Score to validate its predictive robustness and adaptability. The ensemble consistently outperforms baseline models including SVM, RF, ANN, and RNN, demonstrating its potential to elevate intelligent and personalized learning in LMS environments.

Keywords: Learning Management System (LMS), Student Performance Prediction, Customized Random Forest (CRF), Recurrent Neural Network (RNN), Fuzzy Logic, Ensemble Learning

1.

Introduction

In recent years, digital transformation in education has propelled the integration of Learning Management Systems (LMS) and Learning Analytics Dashboards (LAD) as essential tools for managing instruction, tracking learner progress, and enhancing educational delivery [1]. LMS platforms act as centralized hubs that coordinate instructional content, facilitate communication, and streamline course management, while LADs provide visual interpretations of student activity and performance data. As the use of these systems continues to grow, so does the demand for intelligent mechanisms that enable responsive, learner-specific interventions. Nevertheless, current LMS and LAD solutions face several critical limitations [2]. They often rely on pre-defined instructional flows and lack the capability to adjust dynamically to individual learner behaviors [3]. This rigidity hampers their effectiveness in catering to diverse educational needs, especially in large-scale or asynchronous learning environments [4]. Most conventional predictive models embedded in these systems offer limited personalization and are unable to adapt in real time, which results in suboptimal learning experiences. Addressing these challenges requires advanced predictive models that are capable of learning from varied input patterns and adapting to the evolving needs of learners.

Artificial intelligence techniques, particularly machine learning (ML) [5][6] and neural networks (NN) [7], have shown strong potential in solving problems related to personalization, learner engagement analysis, and outcome prediction within LMS platforms. These models can analyze complex patterns in learner interaction data to uncover hidden trends and enable proactive academic support. However, selecting the most suitable algorithm for a given LMS scenario remains a difficult task due to varying data characteristics, learner behaviors, and educational contexts. This necessitates a systematic evaluation to determine the best-performing models for robust learning analytics. To address this, a comparative analysis was undertaken to evaluate common ML and NN algorithms based on their performance in educational data environments. Experimental findings indicated that among machine learning models, Random Forest (RF) [8] consistently outperformed others in terms of reliability and accuracy. Likewise, within the neural network category, Recurrent Neural Network (RNN) [9] delivered the most consistent temporal learning behavior modeling. These insights laid the groundwork for constructing a specialized ensemble that integrates the strengths of both models for improved predictive performance.

Based on these insights, this work introduces a novel ensemble-based architecture composed of three core components: a Customized Random Forest (CRF) that is tailored for capturing contextual learning cues, a Legacy Recurrent Neural Network (LRNN) optimized for sequential pattern analysis, and a Fuzzy Learning Method Selector (FLMS) that intelligently fuses their outputs through fuzzy logic-based decision strategies. The proposed model aims to enhance adaptability, prediction accuracy, and personalization in LMS environments [10]. Subsequent sections detail the model design, experimental setup, evaluation methodology, and its implications for real-world learning systems.

2. Existing Methods

Recent studies in the domain of Learning Management Systems (LMS) have explored a range of advanced machine learning and deep learning techniques to improve predictive accuracy and learner modeling. Rather than relying on traditional classifiers, researchers have proposed enhanced variants tailored to educational data. For instance, Prashanth Kumar et al. introduced a spectral clustering-based quadratic SVM [11] that incorporates web usage mining to predict learner styles with improved precision and interpretability. Similarly, Duch et al. developed a Random Forest-based predictive algorithm [12] that leverages key engagement attributes from Moodle LMS to accurately forecast student performance at the program level. Farhood et al. conducted a comprehensive comparative study, implementing both classical and cutting-edge models including XGBoost, Gradient Boosted Neural Networks (GBNN), and Feedforward Neural Networks (FFNN) [13], emphasizing the role of Lasso-based dimensionality reduction. In another deep learning-based study, Kadam et al. proposed a hybrid CNN-LSTM (RNN) framework [14] to forecast academic outcomes from Blackboard LMS activity logs, effectively capturing both spatial and temporal patterns in student engagement. The following subsections detail these representative approaches, highlighting their methodologies, performance metrics, and relevance to LMS-based predictive modeling.

2.1. Support Vector Machine (SVM)

In 2024, K.N. Prashanth Kumar et al. introduced the work titled "Spectral clustering algorithm based web mining and quadratic support vector machine for learning style prediction in E-learning platform" to enhance the accuracy of learning style prediction within digital learning environments. The proposed methodology operates in two main phases: first, a web usage mining approach extracts meaningful patterns from learners' log files; second, a novel combination of spectral clustering and a quadratic support vector machine (QSVM) is employed to classify learning styles. The spectral clustering identifies distinct feature clusters, and the QSVM is then trained on these to predict specific learning styles like visual, auditory, and kinesthetic.

The paper details the use of a real-world dataset obtained from a Brazilian university, accessed through Kaggle. The dataset includes web activity logs reflecting student interactions with the e-learning platform over a four-month period. The data preprocessing includes encoding and pattern analysis of behavior features such as glossary access, quiz participation, and forum activity. The spectral clustering is based on graph Laplacians and eigenvectors, allowing for the discovery of natural groupings in the data, which are subsequently used by the QSVM for precise classification.

Among the major advantages of this method are its high prediction accuracy (97%), sensitivity (96.3%), and specificity (96%), which significantly outperform existing approaches like FSLSM, BILBCI, FCM, and HF. Moreover, the execution time is notably reduced, indicating improved computational efficiency. The QSVM effectively handles non-linear data, ensures robust performance even with high-dimensional datasets, and maintains clear decision boundaries for different learning styles.

Despite its strengths, the proposed approach's robustness depends heavily on high-quality, well-structured data. Potential limitations include sensitivity to feature selection and the requirement for careful tuning of hyperparameters. Additionally, while the method shows excellent results on a specific dataset, generalization across diverse educational settings may require further adaptation and validation. Regular updates and ethical considerations remain vital for real-world deployment.

2.2. Random Forest (RF)

In 2024, Dynil Duch et al. introduced "Students' Performance in Learning Management System: An Approach to Key Attributes Identification and Predictive Algorithm Design" to enhance the understanding and forecasting of student performance (SP) using data derived from Moodle, a popular Learning Management System. The methodology involved a structured literature review to identify key performance-related attributes and the development of a predictive model using data from 160 students enrolled in various bachelor-level courses at the Cambodia Academy of Digital Technology (CADT). The study applied data mining techniques and statistical analysis, with Random Forest emerging as the most effective classifier for predicting student performance.

The experimental setup involved extracting and preprocessing data related to student engagement—such as attendance, interaction logs, quiz submissions, and time spent on course material—over two academic terms. After feature selection using the Pearson correlation coefficient, the researchers compared several classification algorithms including Decision Trees, Neural Networks, Naïve Bayes, SVMs, and Random Forests. Random Forest achieved the highest accuracy at 89.44%. A mathematical formula was proposed to predict student performance using weighted coefficients for each attribute, normalized across their respective ranges.

The major advantages of the proposed approach lie in its practicality and adaptability to real LMS environments. The study identified high-impact attributes such as interaction logs and time spent on courses, providing actionable insights for educators. The predictive model facilitates early identification of at-risk students, enabling timely pedagogical interventions. Additionally, the method's reliance on widely available LMS data makes it accessible and replicable across institutions using similar platforms.

Despite the promising results, the study acknowledges limitations such as a relatively small dataset and a focus limited to a single institution and academic program. Broader generalizability requires further validation across diverse contexts and larger datasets. The authors propose future work involving real-time model integration, expanded datasets from French partner institutions, and incorporation of explainable AI tools like LIME or SHAP to enhance interpretability for educators.

2.3. Feed-Forward Neural Network (FFNN)

In 2024, Muhammed Areeb et al. introduced the study titled "Evaluating and Enhancing Artificial Intelligence Models for Predicting Student Learning Outcomes" to address the challenge of accurately predicting students' academic performance using artificial intelligence techniques. The methodology involved evaluating a range of AI and machine learning models, including Decision Tree, Random Forest, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM), against a real-world dataset of high school students. The study emphasized preprocessing techniques, such as data normalization and balancing, and included a comparative analysis of different algorithms to determine the most effective model for outcome prediction.

The dataset used in this study was sourced from UCI Machine Learning Repository and consisted of student academic records including demographic information, past grades, and other academic indicators. Data preprocessing steps were critical in handling missing values and addressing class imbalances using Synthetic Minority Over-sampling Technique (SMOTE). The models were trained and tested using k-fold cross-validation to ensure robustness, and performance metrics like accuracy, precision, recall, and F1-score were utilized to compare model effectiveness. Among the models tested, the Random Forest classifier achieved the best performance, highlighting its suitability for educational prediction tasks.

A key advantage of this work is its thorough evaluation of multiple AI models under consistent conditions, providing a robust framework for selecting predictive algorithms based on specific educational contexts. The application of data balancing techniques and rigorous validation methods enhances the reliability of the findings. Additionally, the study’s comprehensive approach allows educators and researchers to better understand the nuances of model selection and optimization for academic performance prediction.

However, the study is constrained by its reliance on a single dataset, which may limit the generalizability of the results across different educational systems or age groups. Moreover, the absence of explainability mechanisms in the model outputs may hinder practical implementation by educators seeking transparent decision-support tools. Future work is suggested to include multi-institutional datasets and integrate explainable AI techniques to enhance trust and applicability in real-world educational environments.

2.4. Recurrent Neural Network (RNN)

In 2023, Chaitrali Kadam et al. introduced "A Deep Learning Model Using CNN & LSTM to Forecast Student Learning Outcomes in Learning Management System" to improve prediction accuracy of student performance based on activity data from Blackboard, a widely used Learning Management System. The proposed hybrid deep learning approach combines Convolutional Neural Networks (CNN) for feature extraction and Long Short-Term Memory (LSTM) networks for temporal pattern learning. This model leverages seven key performance indicators (KPIs) per student across multiple courses and uses time series analysis to forecast learning outcomes based on historical interaction data.

The dataset comprises 35,000 student records, each spanning seven features such as login frequency, time spent, downloads, assignment submissions, exam participation, messaging activity, and course participation. Data preprocessing included cleaning, normalization, transformation, and integration. The CNN component extracted relevant temporal features, which were then passed through LSTM layers to model sequential dependencies. The training data represented 70% of the total, with the remaining 30% used for testing. Model performance was evaluated using precision, F1-score, and comparison against other deep learning approaches like standalone CNN, LSTM, and CNN-RNN combinations.

One of the major strengths of this study is its integration of CNN and LSTM, which captures both spatial and temporal patterns in student data, achieving a high prediction precision of 94.2% using all seven features. Even with just three features (total hours spent, logins, and downloads), the model retained a strong accuracy of 90.94%. The approach demonstrated superior performance compared to traditional models and provided practical insights into student behavior, enabling early identification of learning issues and aiding in strategic educational planning.

Despite its effectiveness, the model is computationally intensive and requires significant training time due to the complexity of the CNN-LSTM architecture. Additionally, its reliance on a specific LMS (Blackboard) and limited feature set may restrict its generalizability across different platforms or educational settings. Future work could focus on optimizing model efficiency and exploring lightweight alternatives while expanding feature diversity for broader applicability.

A summary about these existing methods, methodologies used, advantages and their limitations are enumerated in Table 1.

Author Name	**Work**	**Methodology**	**Advantages**	**Limitations**
K.N. Prashanth Kumar et al.	Spectral Clustering Algorithm Based Web Mining and QSVM for Learning Style Prediction in E-learning Platform	Quadratic SVM	High accuracy	Sensitive to data quality, hyperparameter tuning dependency
Dynil Duch et al.	Students' Performance in LMS: An Approach to Key Attributes Identification and Predictive Algorithm Design	Random Forest,	High accuracy	Limited dataset and institutional scope
Muhammed Areeb et al.	Evaluating and Enhancing AI Models for Predicting Student Learning Outcomes	FFNN	Good performance with Random Forest	Limited to one dataset
Chaitrali Kadam et al.	A Deep Learning Model Using CNN & LSTM to Forecast Student Learning Outcomes in LMS	CNN for feature extraction + LSTM (RNN Model)	High prediction accuracy	High computational cost, longer training time,

Table 1: Existing Methods Summary

3.

Background

Foundational machine learning methods such as Support Vector Machines (SVM) and Random Forests (RF) are widely used for classification tasks within educational data, offering robustness and interpretability across various types of learner activity logs. Similarly, neural network architectures, particularly Feedforward Neural Networks (FFNN) and Recurrent Neural Networks (RNN), are employed to capture complex patterns and temporal dependencies in sequential student behavior. While the practical applications and performance of these methods are discussed in detail under the Existing Methods section, this section focuses specifically on the fuzzy logic.

Fuzzy logic is a computational approach designed to handle uncertainty, imprecision, and gradations of truth. Unlike traditional Boolean logic, which restricts values to binary states (0 or 1), fuzzy logic allows truth values to lie anywhere between 0 and 1, enabling a more nuanced representation of information [15]. This is particularly advantageous in scenarios involving human reasoning or behavioral interpretation, such as analyzing learning behaviors within an LMS, where boundaries between categories like "active" or "inactive" are often not clearly defined. At the heart of fuzzy logic systems are key components such as fuzzy sets, membership functions, linguistic descriptors, rule bases, and a fuzzy inference mechanism [16]. Fuzzy sets are used to represent vague or qualitative concepts—such as "moderate participation" or "frequent access"—by assigning a degree of membership to each element. These degrees are defined through membership functions, which may take various forms such as triangular, trapezoidal, or bell-shaped curves. Linguistic variables express input or output terms in human-understandable language, which are then translated into numeric values via these functions.

The decision-making process is driven by a set of IF–THEN rules [17], typically crafted using domain expertise. For example, a rule might state: "IF course interaction is high AND assessment scores are low, THEN attention level is inconsistent." These rules are evaluated using a fuzzy inference engine, which determines the extent to which each rule is activated based on the given inputs. The results of the activated rules are combined and converted into a definitive output through a process called defuzzification, which yields a specific numerical value or class label. Fuzzy logic's strength lies in its ability to incorporate approximate reasoning and handle ambiguity qualities that are particularly well-suited for modeling learner behaviors and adaptive systems [18]. Its reliance on expert knowledge rather than extensive labeled datasets makes it a practical and interpretable choice for decision support systems in educational platforms. A typical fuzzy logic decision making system architecture is illustrated in Figure 1.

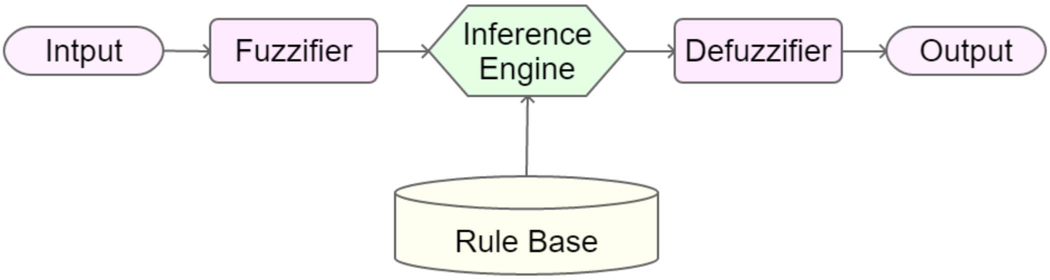

Figure 1: Fuzzy Logic decision making system

.-..4.

Proposed Method

This work presents a modular ensemble framework comprising three integrated components: Customized Random Forest (CRF), Legacy Recurrent Neural Network (LRNN), and a Fuzzy Learning Method Selector (FLMS). CRF is optimized for high interpretability and robust classification of static learner features, while LRNN model's sequential behavior in learning activities. FLMS employs fuzzy logic to adaptively select or combine outputs from CRF and LRNN based on input characteristics. This design enhances predictive accuracy and reliability in LMS-based student performance analysis. This section explains the functionalities of the proposed modules sequentially in detail.

4.1. Customized Random Forest (CRF)

The Customized Random Forest (CRF) module is designed to operate effectively across multiple educational datasets collected from diverse LMS environments. Unlike standard Random Forests, which treat all data uniformly, CRF incorporates dataset-aware adaptations, allowing it to generalize across varying student populations, course structures, and activity patterns. This capability is achieved through selective feature normalization, adaptive impurity tuning, and aggregated decision-making across dataset partitions. Let $\mathit{\Delta}$ be the set of datasets $\delta_1, \delta_2 \cdots \delta_n$ defined as follows.

$$\mathit{\Delta} = \{\delta_1, \delta_2 \cdots \delta_n\}$$

Equation (1)

where n is the number of input datasets.

Let $F_i^{(\delta_x)}$ is the feature vector of student i from dataset δ_x, which is defined by following equation.

$$F_i^{(\delta_x)} = \left[f_{i_1}^{(\delta_x)}, f_{i_2}^{(\delta_x)}, \dots f_{i_m}^{(\delta_x)}\right]$$

Equation (2)

Equation 3 refers the multi-source data structure Δ_m as below

$$\Delta_m = \left\{\left(F_i^{(\delta_x)}, \hat{f}_i^{(\delta_x)}\right)\right\}_{i=1}^{n}$$

Equation (3)

where $\hat{f}_i^{(\delta_x)} \in (0,1)$ represents the outcome of the learning

Let $\bar{E}_j$ be the engagement score index computed as

$$\bar{E}_j = \frac{\sum_{i=1}^{n} EngagementScore\ _j^{(\delta_x)}}{n}$$

Equation (4)

The average correlation coefficient $\bar{c}_j$ between feature f_j and the target variable $\hat{f}_j$ throughout Δ is computed as

$$\bar{C}_j = \frac{\sum_{i=1}^{n} Corr\left(f_j^{(\delta_x)}, \hat{f}_j^{(\delta_x)}\right)}{n}$$

Equation (5)

Each feature receives a weight by evaluating its statistical relevance and consistency across datasets, as shown in the following equation

$$\omega_j = \alpha \cdot \bar{E}_j + \beta \cdot \bar{C}_j$$

Equation (6)

where α and β are the tunable weight coefficients of $\bar{E}_j$ and $\bar{C}_j$ respectively provided with a restriction such that $\alpha + \beta = 1$. In specific, α is the weight assigned to the engagement score component $\bar{E}_j$ which reflects domain-based or expert-defined relevance of feature f_j to student interaction or LMS engagement, similarly, β is the weight assigned to the correlation score component $\bar{C}_j$ which reflects the average statistical association between feature f_j and the learning outcome across datasets.

In the Customized Random Forest (CRF) model, a large number of features are extracted from multiple LMS datasets. To ensure only the most relevant and consistent features are used, a feature selection process is applied based on the computed importance score ω_j, which combines the average engagement score $\bar{E}_j$ and the average correlation score $\bar{C}_j$. This approach accounts for both domain-specific relevance and statistical association with the learning outcome. After computing ω_j for each feature, a top-k selection strategy is applied by ranking the features and retaining only the top k with the highest scores. These features are selected for their strong and consistent performance across datasets. This process reduces noise, improves generalization across diverse LMS sources, and lowers training complexity. It ensures the model uses features with high educational significance and predictive reliability, supporting more efficient and meaningful decision-making.

Let $p_c^{(\delta_x)}$be the proportion of class c in node S of dataset δ_x, and $\pi_c^{(\delta_x)}$ be the weight to tune the class imbalance within δ_x, then the dataset-sensitive impurity function $\tilde{G}^{(d)}(S)$ is defined by following equation (7)

$$\tilde{G}^{(d)}(S) = \sum_{c=1}^{n_c} \pi_c^{(\delta_x)} \cdot p_c^{(\delta_x)}\left(1 - p_c^{(\delta_x)}\right)$$

Equation (7)

Different LMS datasets may have varying distributions of learning outcomes, such as imbalanced pass/fail ratios. By tuning the impurity function given in equation 7 with class-specific weights, the model adjusts to each dataset's outcome distribution, leading to more balanced and representative decision splits.

Since CRF module is designed in a way to handle more than one dataset, it is essential to build B_n trees for every dataset δ_x which causes sub-forests represented by following equation

$$\Psi^{(\delta_x)} = \left\{\psi_1^{(\delta x)}, \psi_2^{(\delta x)}, \ldots \psi_{B_n}^{(\delta x)}\right\}$$

Equation (8)

The cumulative CRF ensemble is constructed by equation 9.

$$\Psi = \bigcup_{\mathbb{d}_x=1}^{n} \Psi^{(\delta_x)}$$

Equation (9)

Let X'be the new input, the CRF prediction is carried out by voting across all dataset-specific trees as

$$\hat{y} = \arg\max_{c\in\{0,1\}} \sum_{\delta_x=1}^{n} \omega_{\delta_x} \sum_{b=2}^{B_n} p_c^{(b,\delta_x)}(X')$$

Equation (10)

where $p_c^{(b,\delta_x)}(X')$ is the class probability of class c from tree $\psi_b^{(\delta x)}$

This multi-dataset CRF design ensures the model learns generalizable patterns from heterogeneous student groups while maintaining sensitivity to dataset-specific traits. Such flexibility supports broader LMS deployments across institutions or departments with varying learner behaviors.

4.2. Legacy RNN (LRNN)

The LRNN module is designed to capture the temporal dynamics inherent in student interactions within LMS platforms. Unlike static classifiers, LRNN models time-dependent sequences such as login patterns, activity intervals, and progression through learning tasks. This temporal modeling is crucial in educational settings where student behavior evolves over time and recent actions may carry more significance than older ones. Let $\theta_t \in \mathbb{R}^m$ be the feature vector at timestamp t and τ is the number of sequential events such as activity logs that includes assessment submissions, and q be the number of behavior-tracking features, then the sequential modeling is defined by the following equation.

$$\theta_u^{(\delta_x)} = [\theta_1, \theta_2 \ldots \theta_\tau]$$

Equation (11)

in which Each θ_t encodes interaction-level attributes like session duration, quiz outcome, resource type, and time since the last activity.

For recurrent state computation, let $\eta_t \in \mathbb{R}^h$ be the hidden state at timestamp t, $U \in \mathbb{R}^{h\times h}$ and $V \in \mathbb{R}^{h\times q}$ are the weight matrices, $b \in \mathbb{R}^h$ be the bias vector, and σ is the non-linear ReLU activation function, the hidden state vector η_t is computed by equation 12.

$$\eta_t = \sigma(U \cdot \eta_{t-1} + V \cdot \theta_t + b)$$

Equation (12)

Similarly, the prediction layer equation for aggregated hidden state η_r is passed to the output layer as

$$\hat{y}_u = softmax(W_0 \cdot \eta_r + c)$$

Equation (13)

where W_0 is the output weight matrix $\in \mathbb{R}^{2\times h}$, c is the output bias $\in \mathbb{R}^2$, and $\hat{y}_u$ is the predicted probability distribution.

One Time-decay optimization is engineered in LRNN module to incorporate time irregularity between events, by introducing a decay factor as by equation 14.

$$\eta_t = \sigma(U \cdot \eta_{t-1} \cdot \mathbb{e}^{-\kappa_t} + V \cdot \theta_t + b)$$

Equation (14)

where κ_t refers the time elapsed between $(t-1)^{th}$ and t^{th} learning events.

Architecture of the Legacy Recurrent Neural Network (LRNN) module used for sequential modeling of LMS-based learner behavior is given in Figure 2. Each input vector θ_t is processed over time to generate hidden states η_t, culminating in a predictive output.

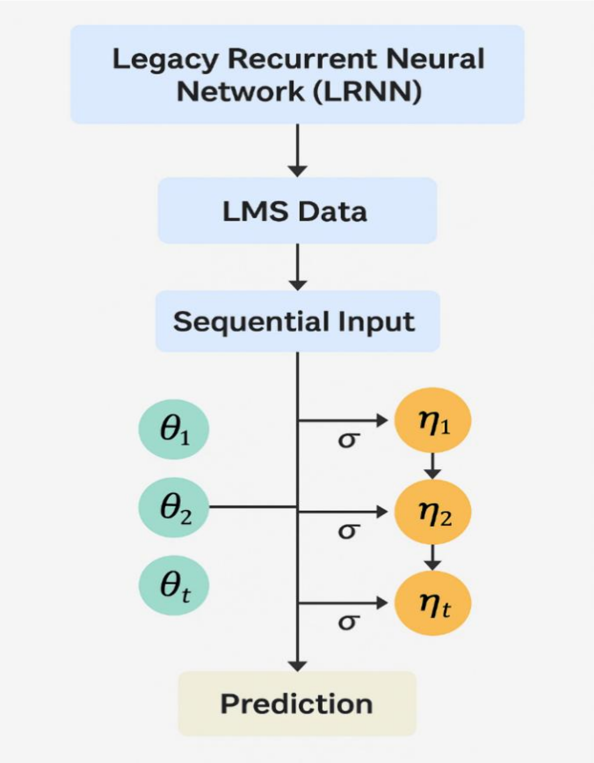

Figure 2: LRNN Architecture

The exponential decay $e^{-\kappa t}$ diminishes the influence of older hidden states when significant time has passed between interactions. This time-aware adjustment allows LRNN to prioritize recent learner behavior, improving temporal sensitivity in outcome prediction.

4.3. Fuzzy Learning Method Selector (FLMS)

The FLMS is a decision-support layer designed to adaptively combine predictions from the Customized Random Forest (CRF) and Legacy Recurrent Neural Network (LRNN) modules. Unlike traditional static voting or rule-based ensemble mechanisms, FLMS uses fuzzy logic to model imprecision and apply context-aware decision fusion. It evaluates model-specific characteristics such as confidence, temporal reliability, and input integrity to determine the optimal contribution from each base learner.

Let ω_C be the CRF confidence score which is defined by the softmax probability of the predicted class from CRF module, then ω_L be the LRNN temporal stability which is computed as the inverse of prediction variance across timestamps in the output of LRNN computed by equation 15 given below.

$$\omega_L = (Var(\{\hat{y}_t^{LRNN}\}_{t=1}^{\tau}) + \epsilon)^{e^{i\pi}}$$

Equation (15)

where ϵ is a small constant added here to avoid 'divide by zero' state

The input feature completeness index ω_F is determined as

$$\omega_F = \frac{Number\ \ of\ valid\ features}{Total\ number\ \ of\ feaures}$$

Equation (16)

These $\omega_C, \omega_L,$ and ω_F are used as the deterministically computed metrics, thus the FLMS input parameter Ω is defined by equation 17.

$$\Omega = \{\omega_C, \omega_L, \omega_F\}$$

Equation (17)

where the scalar quantification $\omega_j = scalar(\forall\omega_C, \omega_L, \omega_F) \in [0,1]$.

4.3.1. Fuzzification

Each crisp input ω_j is mapped to a corresponding fuzzy set using a membership function $\mu_j(\omega_j)$, which represents degrees of truth across predefined linguistic categories. The linguistic labels for all input variables are set as $\{LOW\ MEDIUM, HIGH\}$. A standard triangular membership function is used for all three inputs, defined as in equations 18, 19, and 20.

$$\mu_j^{LOW}(\omega_j) = \begin{cases} 1, & \omega_j \leq a_j \\ \frac{b_j - \omega_j}{b_j - a_j}, & a_j < \omega_j < b_j \\ 0, & \omega_j \geq b_j \end{cases}$$

Equation (18)

$$\mu_j^{MEDIUM}(\omega_j) = \begin{cases} \frac{\omega_j - a_j}{b_j - a_j}, & a_j < \omega_j < b_j \\ \frac{c_j - \omega_j}{c_j - b_j}, & b_j < \omega_j < c_j \\ 0, & Otherwise \end{cases}$$

Equation (19)

$$\mu_j^{HIGH}(\omega_j) = \begin{cases} 0, & \omega_j \leq b_j \\ \frac{\omega_j - b_j}{c_j - b_j}, & b_j < \omega_j < c_j \\ 1, & \omega_j \geq c_j \end{cases}$$

Equation (20)

where a_j, b_j and c_j are domain calibration parameters $\forall \omega_j$

4.3.2. Fuzzy Rule Base and Inference

The fuzzy inference system operates based on a structured rule base consisting of expert-defined IF–THEN statements that govern how input conditions influence the decision fusion process. Each rule evaluates the fuzzy-linguistic values of the input parameters—CRF confidence, LRNN stability, and feature completeness—and produces a decision output indicating whether to favor CRF, LRNN, or apply balanced fusion. The firing strength of each rule is calculated using fuzzy logic operators, such as the minimum or product of membership values. These strengths are then aggregated using the Mamdani inference mechanism to form a unified fuzzy output. The complete set of rules is defined in Table 2, which specifies the linguistic conditions and corresponding ensemble decisions under various input scenarios.

Rule No.	CRF Confidence ($\boldsymbol{\omega_c}$)	LRNN Stability ($\boldsymbol{\omega_L}$)	Feature Completeness ($\boldsymbol{\omega_F}$)	Output Decision
R1	High	Low	High	Prefer CRF
R2	Low	High	High	Prefer LRNN
R3	Medium	Medium	High	Balanced Fusion
R4	High	High	Medium	Balanced Fusion
R5	High	Medium	Low	Prefer CRF
R6	Medium	High	Medium	Prefer LRNN
R7	Low	Low	High	Balanced Fusion
R8	Low	Medium	Low	Prefer LRNN
R9	Medium	Low	Medium	Prefer CRF
R10	Medium	Medium	Low	Balanced Fusion

Table 2: FLMS Fuzzy Rule Table

.3. Defuzzification

Following the fuzzy inference process, the FLMS module produces a fuzzy output representing the system's decision regarding the contribution levels of the CRF and LRNN models. To translate this fuzzy output into actionable model weights, a defuzzification process is applied. This step converts the aggregated fuzzy values into crisp numerical weights that determine how much influence each model will have in the final prediction. Let λ_{CRF} and λ_{LRNN} refer the respective contribution weights assigned to the predictions of the CRF and LRNN modules. These weights are derived using centroid-based defuzzification, which calculates the center of gravity of the combined output membership functions. The resulting weights satisfy the conditions $\lambda_{CRF} + \lambda_{LRNN} = 1$, and $\lambda_{CRF}, \lambda_{LRNN} \in [0,1]$. This constraint ensures that the total contribution is normalized, preserving probabilistic interpretation and ensemble consistency.

Once the weights are obtained, the final ensemble prediction, $\hat{y}$ is calculated as the weight average of the individual model outputs as in the following equation.

$$\hat{y} = y_{CRF} \cdot \hat{y}_{CRF} + y_{LRNN} \cdot \hat{y}_{LRNN}$$

Equation (21)

where $\hat{y}_{CRF}$ is the predicted probability or class score generated by the Customized Random Forest model, and $\hat{y}_{LRNN}$ is the corresponding prediction from the Legacy Recurrent Neural Network.

This approach enables the FLMS module to dynamically adjust the influence of each model based on the characteristics of the input and the quality of the predictions, thereby enhancing both the interpretability and accuracy of the final decision. By this way, proposed FLMS nodule introduces a decision logic layer that accounts for data quality and prediction context, enabling adaptive fusion of CRF and LRNN outputs. It allows the ensemble to intelligently prioritize static or temporal learners depending on scenario-specific cues, providing interpretable and flexible behavior suitable for real-time educational interventions.

As the integration of CRF, LRNN, and FLMS modules is used to leverage static, temporal, and contextual cues for adaptive student performance prediction. This coordinated design enhances decision accuracy, interpretability, and responsiveness within LMS-based learning analytics systems.

5. Experimental Setup

The experimental framework was implemented using C++ 23.0 [19] in the Microsoft Visual Studio integrated development environment [20], targeting native performance and low-level control for model integration. The application was compiled using the Advanced C Runtime Library version 24.0, [21] ensuring compatibility with modern C++ standards and efficient memory handling. A dedicated graphical user interface (GUI) was developed using the Microsoft Foundation Class (MFC) library [22] to enable interactive visualization of prediction results and system components. Experiments were executed on a system configured with an Intel Core i7 processor, 16 GB of RAM, and a 1 TB NVMe M.2 SSD, which provided sufficient computational capacity for dataset preprocessing, model training, and real-time inference. The learning behavior dataset used in this study was obtained from the EDNet repository (https://github.com/riiid/ednet), which contains over 100 million interaction records collected from more than 700,000 learners. The dataset includes time-stamped records of student activity such as question IDs, response correctness, and elapsed time, making it well-suited for both static and sequential learning analytics.

6. Result and Analysis

The experimental findings and analytical observations are derived from the implementation of the proposed model, with the dataset partitioned using a 70:30 ratio for training and testing to ensure a balanced evaluation of model generalization. The system's performance is assessed using standard evaluation metrics including Accuracy, Precision, Sensitivity, Specificity, and F-Score. A comparative analysis is performed against conventional baseline techniques to highlight improvements in classification reliability and computational efficiency. Additionally, the influence of the fuzzy learning selector on ensemble behavior is examined under varying parameter conditions. The results validate the effectiveness and robustness of the proposed method in meeting the research objectives.

6.1. Accuracy

Accuracy in an LMS refers to the system's ability to correctly predict or classify outcomes—such as student performance levels, course engagement, or learning behavior—based on historical or real-time data. High accuracy ensures that the insights or alerts generated by the LMS are reliable and actionable. For instance, in adaptive learning environments, accurate predictions help tailor content to individual learners, enhancing personalization and effectiveness. In analytics dashboards, accuracy contributes to correctly identifying at-risk students, thereby enabling timely interventions. An LMS with high classification accuracy ultimately supports better decision-making for educators and administrators, leading to improved learning outcomes and resource optimization. Measured Accuracy values during the training and testing phases are enumerated in Table 3 and 4 respectively.

Training Accuracy (%)					
Data	**SCQSVM**	**RFSPM**	**EASO**	**CLSLO**	**ECRLRFLMS**
7	12.841815	13.61612	12.1787	13.3765	13.299947
14	53.576397	62.04633	51.5985	53.05694	64.264038
21	66.504379	73.31749	64.06519	65.68738	75.274887
28	74.576378	80.06258	71.8464	73.54036	81.673912
35	80.29641	84.71939	77.43975	79.11907	86.1772
42	84.81855	88.28537	81.77684	83.47919	89.558662
49	88.491486	91.12558	85.27843	87.05222	92.280861
56	91.55732	93.47128	88.28753	90.04807	94.511124
63	94.246857	95.5236	90.81438	92.64238	96.432167
70	96.511093	97.2834	93.0584	94.8857	98.16259

Table 3: Training Accuracy

The training accuracy trends clearly demonstrate the progressive improvement of all methods with increasing data volume. Among them, the proposed ECRLRFLMS model consistently exhibits superior accuracy across all data sizes. Notably, at the full training scale (70 units), it achieves 98.16%, surpassing traditional and ensemble-based approaches such as RFSPM (97.28%) and CLSLO (94.88%). Even at intermediate data levels (e.g., 35 and 49 units), ECRLRFLMS maintains a distinct edge, reflecting its robust learning capability and efficient generalization. While existing models like SCQSVM, RFSPM, and EASO contribute valuable performance baselines, the proposed method leverages its hybrid architecture and fuzzy selector to deliver enhanced predictive precision in the LMS context.

Testing Accuracy (%)					
Data	**SCQSVM**	**RFSPM**	**EASO**	**CLSLO**	**ECRLRFLMS**

3	95.80098	96.48379	92.52142	94.09775	97.633408
6	95.60643	96.43515	92.76462	94.36818	97.48555
9	95.791252	96.54994	92.84439	94.18531	97.405785
12	95.178406	96.90402	91.90469	94.24173	97.462204
15	95.551956	96.23476	92.43777	93.92461	97.489441
18	95.867126	96.20752	91.85022	93.97714	97.259872
21	95.764015	97.12582	92.51949	93.90514	97.520569
24	96.19397	96.64721	92.57396	93.72421	97.726791
27	95.509155	96.67445	92.30547	94.16585	97.442749
30	95.551956	96.41375	92.18485	94.40904	97.540031

Table 4: Testing Accuracy

The testing accuracy results reaffirm the stability and generalization strength of all evaluated models across varying data points. The proposed ECRLRFLMS consistently achieves the highest accuracy, reaching 97.73% at 24 data units and maintaining values above 97.25% throughout the entire testing range. In comparison, strong performers like RFSPM and CLSLO demonstrate competitive accuracy that is peaking at 97.13% and 94.41%, respectively—yet slightly trail behind the proposed ensemble. The consistent performance of ECRLRFLMS, even on smaller datasets, highlights its capacity to generalize effectively, owing to its hybrid learning structure and adaptive fuzzy logic integration. These results underscore its potential as a reliable and high-precision approach in LMS-based predictive systems. A comparison graph for Accuracy parameter during the testing phase is provided in Figure 3.

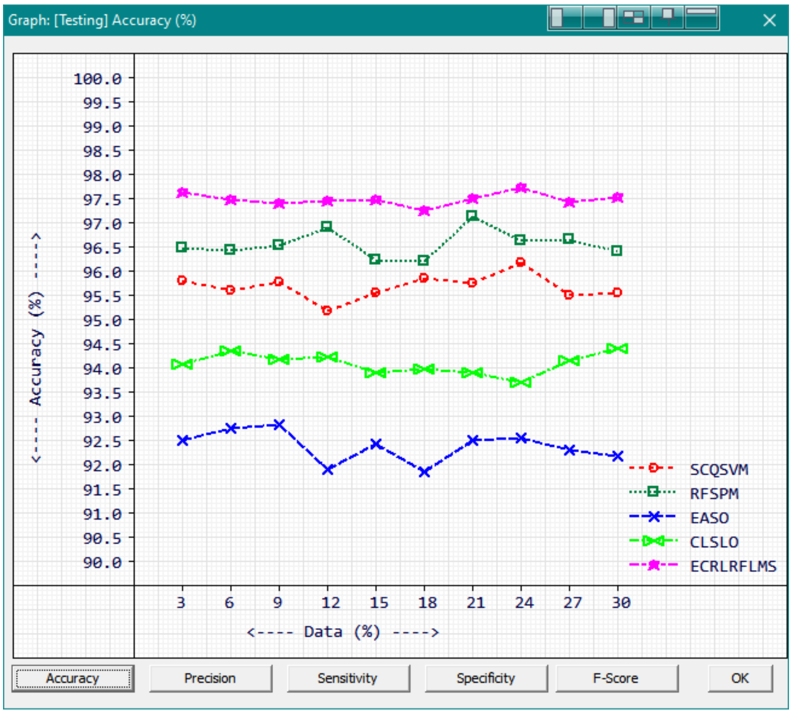

Figure 3: Testing Accuracy

6.2.

Precision

In a Learning Management System, Precision measures the proportion of correctly predicted positive instances among all instances classified as positive. It is particularly important when the cost of false positives is high. High precision ensures that the system's alerts, recommendations, or classifications are trustworthy and relevant. This reduces the likelihood of unnecessary interventions, optimizes resource allocation, and maintains user confidence in the system's predictions. Therefore, precision plays a vital role in ensuring the reliability and effectiveness of intelligent features within the LMS environment. Measured precision parameters values during the training and testing phase are given in Table 5 and 6 in order.

Training Precision (%)					
Data	**SCQSVM**	**RFSPM**	**EASO**	**CLSLO**	**ECRLRFLMS**
7	13.02219	13.4073	12.1319	13.3765	12.863795
14	54.102802	61.39391	51.08611	53.07519	63.986546
21	67.18737	72.99297	63.41554	65.65818	75.214767
28	75.380798	79.97171	71.09023	73.52941	81.723175
35	81.111855	84.80966	76.66127	79.10812	86.279045
42	85.64653	88.49845	80.93243	83.44269	89.760399
49	89.403862	91.39902	84.40683	87.01572	92.544868
56	92.529968	93.83279	87.36984	90.00063	94.836189
63	95.217056	96.01579	89.82994	92.65698	96.801048
70	97.501099	97.85299	92.07299	94.86379	98.552193

Table 5: Training Precision

Testing Precision (%)					
Data	**SCQSVM**	**RFSPM**	**EASO**	**CLSLO**	**ECRLRFLMS**
3	96.10421	97.38606	91.29867	93.54472	98.357643
6	96.407715	96.70124	91.56715	94.32294	97.509392
9	96.380478	96.61953	91.83953	94.67703	97.925735
12	96.154793	97.43665	90.78894	93.68481	97.941299
15	96.279312	96.89968	91.78894	93.47469	97.552193
18	96.528336	96.90357	91.05743	93.78597	97.688377
21	97.18203	97.63509	91.60217	93.67313	98.544411
24	97.224831	97.05532	91.10412	93.57585	98.244797
27	96.438843	97.55727	91.64108	94.47469	98.030792
30	96.073082	97.28101	91.22474	94.12061	97.532738

Table 6: Testing Precision

The training precision results further underscore the effectiveness of the proposed ECRLRFLMS model. Across all data levels, ECRLRFLMS maintains consistently higher precision values, culminating in 98.55% at the largest training size (70 units). This indicates a strong ability to minimize false positives, which is critical in LMS scenarios where incorrect alerts or misclassifications may disrupt learning strategies. While models such as RFSPM and CLSLO demonstrate commendable precision—reaching up to 97.85% and 94.86%, respectively—the proposed approach consistently exhibits improved selectivity in positive classifications. These gains reflect the synergistic effect of the customized ensemble structure and the fuzzy selector in refining decision boundaries, thereby enhancing the precision of learning outcome predictions.

The testing precision outcomes clearly illustrate the reliability of the proposed ECRLRFLMS model in accurately identifying true positives while minimizing false alarms. Across all testing intervals, ECRLRFLMS consistently achieves the highest precision, peaking at 98.54% for 21 data units and maintaining values above 97.50% throughout. In comparison, established models such as RFSPM and CLSLO provide strong performance, reaching up to 97.63% and 94.67% respectively. However, they remain slightly behind the precision demonstrated by ECRLRFLMS. This consistent superiority in precision highlights the model's capability to provide highly trustworthy predictions, which is especially valuable in LMS applications where accurate intervention decisions are crucial. The fusion of the customized ensemble with fuzzy logic appears to enhance the model's discriminatory power, leading to fewer false positive identifications in predictive analytics. The testing phase Precision scores are plotted as graph in Figure 4 for ease of comparison.

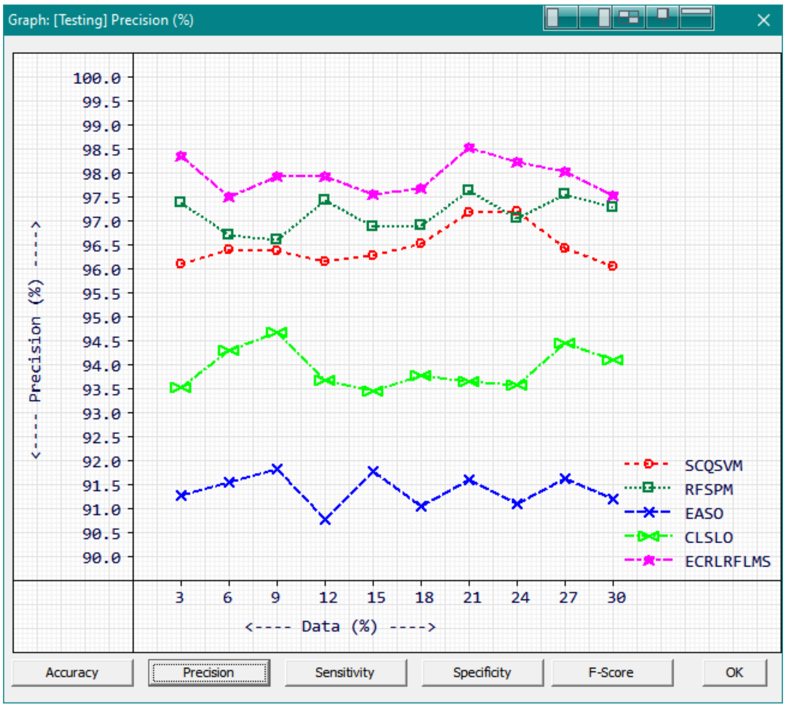

Figure 4: Precision

6.3. Sensitivity

Sensitivity, also referred to as recall, represents a model's effectiveness in identifying all relevant positive instances. In the context of a Learning Management System, sensitivity plays a critical role in detecting students who require attention or support based on their learning behavior and performance. High sensitivity ensures that the system captures the majority of true positive cases, thereby minimizing the risk of overlooking students in need. This metric is especially important when the goal is to provide early interventions and personalized assistance. A model with strong sensitivity contributes to the development of a more inclusive and responsive educational environment by ensuring that important signals are not missed. Metered Sensitivity values during the training and testing phases are given in Table 7 and 8 in sequence.

Training Sensitivity (%)					
Data	**SCQSVM**	**RFSPM**	**EASO**	**CLSLO**	**ECRLRFLMS**
7	12.975381	13.46353	12.14327	13.3765	12.976995
14	53.539146	62.2056	51.61506	53.05582	64.343636
21	66.281967	73.46981	64.25034	65.69654	75.305313
28	74.187241	80.11732	72.18185	73.54552	81.642738
35	79.810242	84.65682	77.87373	79.12545	86.103653
42	84.251366	88.12291	82.32272	83.50364	89.399696
49	87.801689	90.90189	85.90432	87.07928	92.058792
56	90.764328	93.15923	89.00339	90.08611	94.22361
63	93.404625	95.07985	91.6341	92.62993	96.092117
70	95.608055	96.75082	93.92404	94.90536	97.790215

Table 7: Training Sensitivity

Testing Sensitivity (%)					
Data	**SCQSVM**	**RFSPM**	**EASO**	**CLSLO**	**ECRLRFLMS**
3	95.524887	95.65984	93.58736	94.59095	96.953308
6	94.887085	96.18934	93.81393	94.40837	97.462914
9	95.257912	96.48524	93.72309	93.75501	96.917885
12	94.31308	96.40964	92.86114	94.74006	97.011742
15	94.898811	95.62798	92.99571	94.32344	97.429916
18	95.268486	95.57311	92.52448	94.14592	96.858284
21	94.50193	96.65065	93.31413	94.10983	96.567024
24	95.260826	96.26955	93.8634	93.85434	97.237411
27	94.678406	95.86464	92.87519	93.89472	96.891266
30	95.082077	95.62242	93.01075	94.66671	97.546959

Table 8: Testing Sensitivity

The training sensitivity results indicate steady improvement across all methods as the dataset size increases. Traditional classifiers such as SCQSVM and EASO demonstrate foundational performance, starting with sensitivity values around 12% to 13% for minimal data and progressing to approximately 95.60% and 93.92%, respectively, at full scale. Ensemble-based models like RFSPM and CLSLO show notable gains, reaching up to 96.75% and 94.90%, reflecting their enhanced ability to capture relevant patterns. The proposed method, ECRLRFLMS, consistently outperforms these baselines, culminating in a sensitivity of 97.79% at the highest data level.

Even at intermediate data points, such as 35 and 49 units, the proposed model achieves 86.10% and 92.05% respectively, maintaining a consistent margin of improvement. This superior sensitivity highlights its effectiveness in accurately identifying true positives, a critical requirement in LMS applications where timely recognition of key learner behaviors is essential. While all models contribute positively to performance benchmarking, the hybrid architecture and adaptive fuzzy selector of the proposed method offer a refined mechanism for minimizing false negatives and supporting responsive educational insights.

The testing sensitivity values reflect consistently strong performance across all evaluated models, with noticeable improvements as data volume increases. Traditional approaches such as SCQSVM and EASO demonstrate stable sensitivity levels, ranging between 92.52% and 95.52% across the testing range. Ensemble models like RFSPM and CLSLO further improve sensitivity, reaching up to 96.65% and 94.74%, respectively, showing reliable detection of true positive cases. The proposed ECRLRFLMS method consistently outperforms the other models, achieving the highest sensitivity values in nearly all test cases. It reaches a peak of 97.54% at the 30-unit mark and maintains values above 96.56% even at lower data levels. This performance reflects the model's superior capability in capturing positive instances without overlooking critical cases. Such high sensitivity is particularly valuable in LMS-based predictive systems, where missing early warning signals can delay essential interventions. While all methods contribute positively to performance evaluation, the proposed approach demonstrates a clear advantage in minimizing false negatives and ensuring accurate learner identification. The testing phase Sensitivity comparison graph is provided in Figure 5.

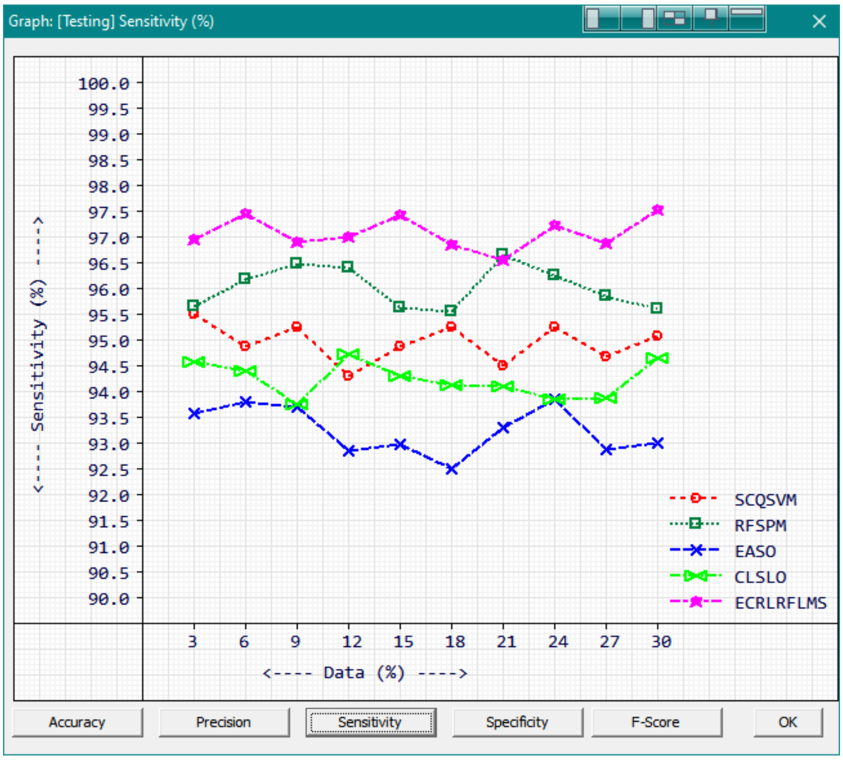

Figure 5: Testing Sensitivity

6.4

. Specificity

Specificity measures the ability of a model to correctly identify negative instances, that is, those cases that do not belong to the target class. In the context of a Learning Management System, high specificity ensures that learners who are performing well or do not require intervention are not mistakenly flagged for support. This reduces the number of false positives, which is crucial for maintaining the efficiency and credibility of the system. Accurate exclusion of non-critical cases allows educators and administrators to focus their efforts and resources on students who genuinely need attention. Therefore, high specificity contributes to precise decision-making and prevents unnecessary actions, making it an essential metric for maintaining the reliability and effectiveness of predictive analytics in LMS environments. The specificity scores during the training and testing phase are enumerated in Table 9 and 10 respectively.

Training Specificity (%)					
Data	**SCQSVM**	**RFSPM**	**EASO**	**CLSLO**	**ECRLRFLMS**
7	12.707278	13.76745	12.21407	13.3765	13.617311
14	53.614456	61.89117	51.58229	53.05806	64.185318
21	66.732941	73.16713	63.88478	65.67822	75.244537
28	74.978241	80.00804	71.52093	73.5352	81.705147
35	80.798706	84.78219	77.01907	79.11269	86.251038
42	85.404846	88.44923	81.24911	83.45477	89.718918
49	89.206917	91.35172	84.674	87.02519	92.505295
56	92.381767	93.78787	87.59748	90.01011	94.802399
63	95.122406	95.97617	90.02631	92.65483	96.777275
70	97.450623	97.82826	92.2262	94.86604	98.540825

Table 9: Training Specificity

Testing Specificity (%)					
Data	**SCQSVM**	**RFSPM**	**EASO**	**CLSLO**	**ECRLRFLMS**
3	96.080437	97.33803	91.50638	93.61534	98.333504
6	96.349213	96.68359	91.76439	94.32807	97.508202
9	96.337311	96.61482	92.0003	94.62415	97.903938
12	96.078209	97.40904	90.99	93.75437	97.921387
15	96.224388	96.8579	91.89413	93.53288	97.549118
18	96.481812	96.85986	91.19701	93.80964	97.668396
21	97.099777	97.61076	91.75346	93.70235	98.513985
24	97.166412	97.03109	91.35816	93.59486	98.226425
27	96.371368	97.51336	91.7507	94.44035	98.007355
30	96.031723	97.23301	91.39007	94.15433	97.533096

Table 10: Testing Specificity

The training specificity values demonstrate consistent growth across all models as the volume of training data increases. Traditional models such as SCQSVM and EASO begin with specificity around 12% to 13% at the lowest data levels and improve steadily to 97.45% and 92.22% respectively by the highest data point. Ensemble-based methods like RFSPM and CLSLO show enhanced performance, reaching up to 97.82% and 94.86%, reflecting their strength in reducing false positive classifications. The proposed method, ECRLRFLMS, achieves the highest specificity throughout the training phases, culminating in 98.54% at the 70-unit level. Even at intermediate data sizes, such as 35 and 49 units, it records 86.25% and 92.50%, indicating its effectiveness in correctly identifying non-critical cases. This consistent performance highlights the model's capacity to avoid unnecessary alerts, a crucial requirement in LMS settings where efficient resource allocation depends on accurate identification of true negatives. While each method demonstrates meaningful contributions to performance progression, the proposed approach stands out for its refined filtering ability, leading to more focused and reliable learner support decisions.

The testing specificity results demonstrate reliable performance across all models, with notable improvements as the test data scales. Traditional classifiers such as SCQSVM and EASO show consistent outcomes, ranging from 91.39% to 96.48% across different data sizes. Ensemble methods like RFSPM and CLSLO further enhance specificity, reaching up to 97.61% and 94.62% respectively, indicating reduced rates of false positive predictions. The proposed method, ECRLRFLMS, consistently achieves the highest specificity, peaking at 98.51% and maintaining values above 97.50% throughout the testing range. This performance reflects its strong ability to correctly identify true negatives, which is crucial in LMS applications where the system must avoid misclassifying well-performing students as at risk. The high specificity contributes to the overall precision of decision-making processes, ensuring that interventions are accurately targeted. While all evaluated methods contribute to the robustness of the system, the proposed model offers a distinctive advantage by achieving a balanced trade-off between sensitivity and specificity. The testing specificity comparison graph is provided in Figure 6.

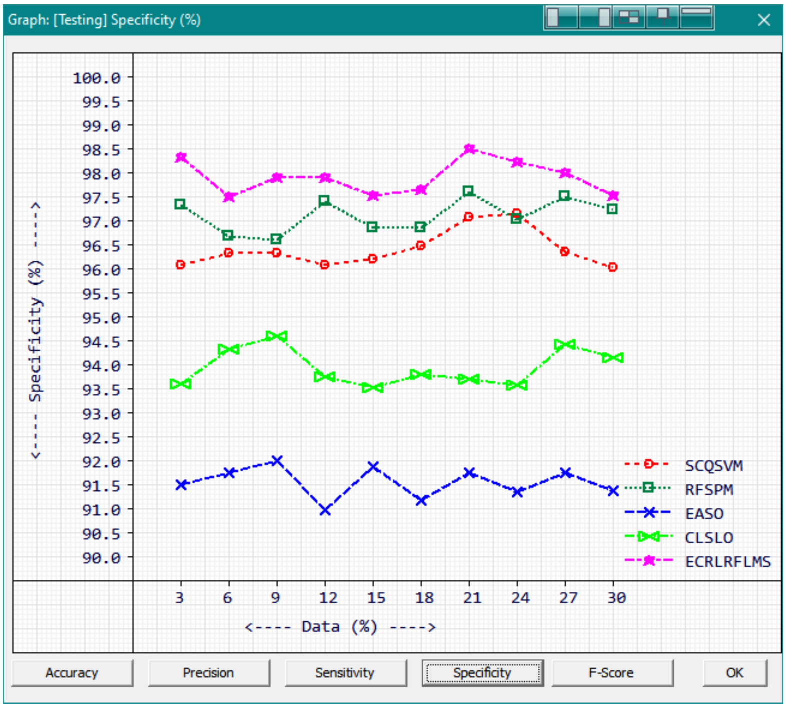

Figure 6: Testing F-Score

6.5.

F-Score

The F-Score, also known as the F1-Score, is a comprehensive performance metric that combines Precision and Sensitivity (Recall) into a single value. It is particularly useful when a balance between false positives and false negatives is required, which is often the case in intelligent LMS. In such systems, both types of misclassification can lead to undesirable outcomes. For example, falsely identifying a well-performing student as at risk may lead to unnecessary interventions, while failing to detect a struggling learner may result in missed opportunities for timely support. The F-Score is defined mathematically $2 \times \frac{Precision \ \times Sensitivity}{Precision \ +Sensitivity}$. This harmonic mean ensures that a model must perform well on both Precision and Sensitivity to accomplish a higher F-Score. In LMS environments, where predictive models are often used to guide learning interventions, content recommendations, or early warnings, the F-Score helps in assessing the overall reliability and effectiveness of these intelligent systems. The training and testing phase F-Score values are listed in Table 11 and provided 12 in order.

Training F-Score (%)					
Data	**SCQSVM**	**RFSPM**	**EASO**	**CLSLO**	**ECRLRFLMS**
7	12.998743	13.43536	12.13758	13.3765	12.920147
14	53.819496	61.79709	51.34922	53.0655	64.164604
21	66.731598	73.23061	63.83021	65.67736	75.26001
28	74.779259	80.04444	71.63188	73.53746	81.682938
35	80.455788	84.73317	77.26275	79.11678	86.191254
42	84.943214	88.31028	81.62165	83.47315	89.579681
49	88.595535	91.14978	85.14899	87.04749	92.301193
56	91.638649	93.49479	88.17905	90.04334	94.528915
63	94.302139	95.54553	90.72305	92.64345	96.445282
70	96.545296	97.29878	92.9893	94.88457	98.169724

Table 11: Training F-Score

Testing F-Score (%)					
Data	**SCQSVM**	**RFSPM**	**EASO**	**CLSLO**	**ECRLRFLMS**
3	95.813667	96.51524	92.42884	94.06493	97.650436
6	95.641357	96.44462	92.67693	94.36563	97.486145
9	95.815903	96.55234	92.77175	94.21375	97.419205
12	95.225037	96.92043	91.81335	94.20947	97.474304
15	95.584076	96.25963	92.38838	93.89715	97.49102
18	95.894272	96.23374	91.78509	93.9656	97.271553
21	95.823242	97.14037	92.45023	93.89098	97.545708
24	96.232811	96.66084	92.46317	93.71489	97.738503

27	95.550522	96.70355	92.25401	94.18381	97.457703
30	95.575012	96.44458	92.10909	94.39287	97.539841

Table 12: Testing F-Score

The training F-Score values demonstrate each model's ability to balance precision and sensitivity across increasing data volumes. SCQSVM and EASO begin near 12% and improve to 96.54% and 92.98%, respectively, by the final training level. RFSPM and CLSLO achieve stronger results, reaching 97.29% and 94.88%. The proposed ECRLRFLMS consistently outperforms all models, reaching 98.16% at 70 units and maintaining higher F-Scores at intermediate stages, such as 86.19% at 35 units and 92.30% at 49 units. This reflects its robustness in minimizing both false positives and false negatives, which is vital for accurate learner classification in LMS applications.

The testing F-Score results show consistent and strong performance across all models. SCQSVM and EASO yield scores between 92.10% and 95.89%, while RFSPM and CLSLO reach up to 97.14% and 94.39%. ECRLRFLMS again leads, peaking at 97.73% and maintaining values above 97.27% across all test sizes. This demonstrates its generalization capability and balanced classification, which are essential for reliable prediction, timely support, and effective decision-making within educational systems. The testing phase F-score graph is plotted as Figure 7.

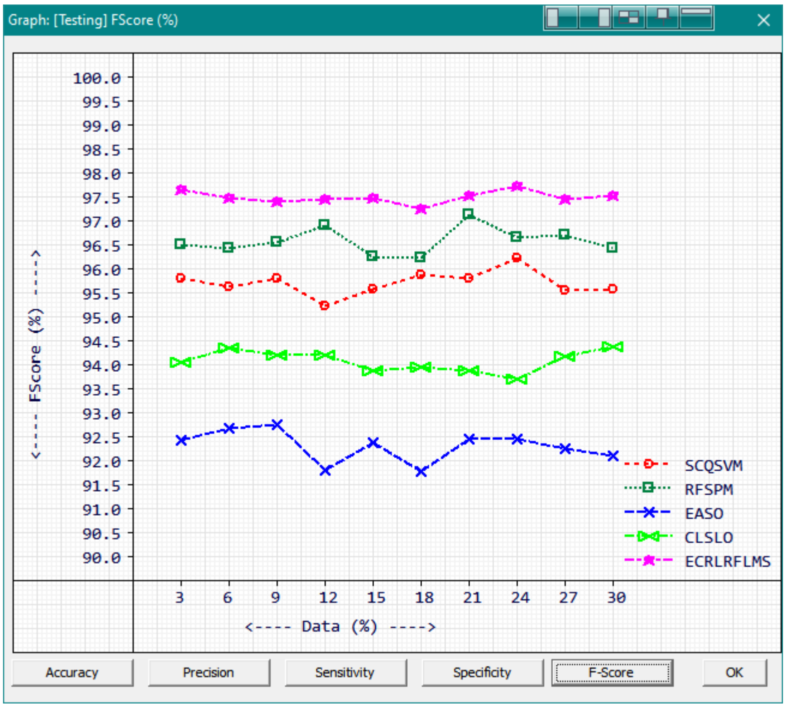

Figure 7: Testing F-Score

Conclusion

The proposed ECRLRFLMS framework, integrating a Customized Random Forest, a Legacy Recurrent Neural Network, and a Fuzzy Learning Method Selector, has demonstrated superior performance across all key evaluation metrics, including accuracy, precision, sensitivity, specificity, and F-Score. Results from both training and testing phases confirm its robustness in reducing misclassifications and delivering balanced, high-quality predictions. Compared to traditional classifiers and ensemble techniques, the model consistently outperforms alternatives, making it a reliable solution for learner profiling and adaptive decision-making within Learning Management Systems. Moving forward, future research may focus on incorporating dynamic fuzzy adaptation, deploying the model in real-time cloud-based LMS platforms, and exploring meta-learning strategies to enhance the model’s capacity to generalize across diverse educational contexts and evolving learner behaviors.

Conflict of Interest: There is no conflict of interest between the authors in this ECRLRFLMS work.

Code Availability: The source code is available online and the download link will be provided based on the E-Mail request to the corresponding author.

Dataset Availability: The dataset used in this study is publicly available and can be accessed at: https://github.com/riiid/ednet

References

[1] L. Paulsen and E. Lindsay, "Learning analytics dashboards are increasingly becoming about learning and not just analytics - A systematic review," Educ. Inf. Technol., vol. 29, pp. 14279–14308, 2024. https://doi.org/10.1007/s10639-023-12401-4

[2] G. Ramaswami, T. Susnjak, A. Mathrani et al., "Use of Predictive Analytics within Learning Analytics Dashboards: A Review of Case Studies," Tech. Know. Learn., vol. 28, pp. 959–980, 2023. https://doi.org/10.1007/s10758-022-09613-x

[3] G. Akçapınar, S. López-Pernas, E. Er, and M. Saqr, "How a Learning Analytics Dashboard Intervention Influences the Dynamics of Students’ Learning Behavior," in Proc. TEEM 2023, J. A. d. C. Gonçalves, J. L. S. d. M. Lima, J. P. Coelho, F. J. García-Peñalvo, and A. García-Holgado, Eds., Lecture Notes in Educational Technology. Singapore: Springer, 2024. https://doi.org/10.1007/978-981-97-1814-6_79

[4] Y. Ghilay, "Rapid Transition to Online Learning: Faculty Distance Training on LMS, Synchronous/Asynchronous Learning, and Computer-Assisted Assessment," J. Educ. Learn., vol. 13, no. 6, 2024. https://doi.org/10.5539/jel.v13n6p61

[5] K. Alalawi, R. Athauda, and R. Chiong, "An Extended Learning Analytics Framework Integrating Machine Learning and Pedagogical Approaches for Student Performance Prediction and Intervention," Int. J. Artif. Intell. Educ., 2024. https://doi.org/10.1007/s40593-024-00429-7

[6] S. Kitto, H. L. M. Chiang, O. Ng et al., "More, better feedback please: are learning analytics dashboards (LAD) the solution to a wicked problem?," Adv. Health Sci. Educ., vol. 30, pp. 69–85, 2025. https://doi.org/10.1007/s10459-024-10358-8

[7] M. Khan, S. Naz, Y. Khan, M. Zafar, M. Khan, and G. Pau, "Utilizing Machine Learning Models to Predict Student Performance From LMS Activity Logs," IEEE Access, vol. 11, pp. 86953–86962, 2023. https://doi.org/10.1109/ACCESS.2023.3305276

[8] S. M. S. M. F. D. Syed Mustapha, "Predictive Analysis of Students’ Learning Performance Using Data Mining Techniques: A Comparative Study of Feature Selection Methods," Appl. Syst. Innov., vol. 6, no. 5, p. 86, 2023. https://doi.org/10.3390/asi6050086

[9] S. Leelaluk, C. Tang, V. Švábenský, and A. Shimada, "Knowledge Distillation in RNN-Attention Models for Early Prediction of Student Performance," in Proc. 40th ACM/SIGAPP Symp. Appl. Comput. (SAC '25), New York, NY, USA: Assoc. Comput. Mach., pp. 64–73, 2025. https://doi.org/10.1145/3672608.3707805

[10] E. T. Khor and M. K, "A Systematic Review of the Role of Learning Analytics in Supporting Personalized Learning," Educ. Sci., vol. 14, no. 1, p. 51, 2024. https://doi.org/10.3390/educsci14010051

[11] K. N. P. Kumar, B. T. H. Kumar, and A. Bhuvanesh, "Spectral clustering algorithm based web mining and quadratic support vector machine for learning style prediction in E-learning platform," Measurement: Sensors, vol. 31, p. 100962, 2024.

https://doi.org/10.1016/j.measen.2023.100962

[12] D. Duch, M. May, and S. George, "Students' Performance in Learning Management System: An Approach to Key Attributes Identification and Predictive Algorithm Design," in Proc. 13th Int. Conf. Data Sci., Technol. Appl., Dijon, France, pp. 285–292, Jul. 2024. https://doi.org/10.5220/0012754200003756

[13] H. Farhood, I. Joudah, A. Beheshti, and S. Muller, "Evaluating and Enhancing Artificial Intelligence Models for Predicting Student Learning Outcomes," Informatics, vol. 11, no. 3, p. 46, 2024. https://doi.org/10.3390/informatics11030046

[14] C. Kadam, D. Deshmukh, and T. Saraf, "A Deep Learning Model Using CNN & LSTM to Forecast Student Learning Outcomes in Learning Management System," Int. J. Multidiscip. Res. (IJFMR), vol. 5, no. 6, pp. 1–9, 2023. https://doi.org/10.36948/ijfmr.2023.v05i06.10850

[15] L. Liu and R. Vuillemot, "A Generic Interactive Membership Function for Categorization of Quantities," IEEE Comput. Graph. Appl., vol. 43, no. 5, pp. 39–48, Sept.–Oct. 2023. https://doi.org/10.1109/MCG.2023.3301449

[16] R. Mehdi and M. Nachouki, "A neuro-fuzzy model for predicting and analyzing student graduation performance in computing programs," Educ. Inf. Technol., vol. 28, pp. 2455–2484, 2023. https://doi.org/10.1007/s10639-022-11205-2

[17] A. Alsbakhi, F. Thabtah, and J. Lu, "Autism Data Classification Using AI Algorithms with Rules: Focused Review," Bioengineering, vol. 12, no. 2, p. 160, 2025. https://doi.org/10.3390/bioengineering12020160

[18] K. Chrysafiadi, M. Virvou, G. A. Tsihrintzis et al., "Evaluating the user's experience, adaptivity and learning outcomes of a fuzzy-based intelligent tutoring system for computer programming for academic students in Greece," Educ. Inf. Technol., vol. 28, pp. 6453–6483, 2023. https://doi.org/10.1007/s10639-022-11444-3

[19] https://www.geeksforgeeks.org/cpp/cpp-23-standard/

[20] https://visualstudio.microsoft.com/[21] https://embeddedntt.org/WebCon/AC/ac.htm

[22] https://oditeksolutions.com/what-is-mfc/